In the Claims

Claims 1-46 remain pending in this application. No claims have been presently amended. No claims have been canceled. New claims 41-46 have been added.

1. (Original) A computerized method of virtualizing graphics resources comprising:

receiving, by a graphics kernel, an allocation request for a graphics resource from a graphics client;

allocating, by the graphics kernel, the graphics resource to the graphics client;

returning, by the graphics kernel, an address for the graphics resource to the graphics client;

receiving, by the graphics kernel, a command from the graphics client specifying the address; and

managing, by the graphics kernel, the graphics resource.

2. (Original) The computerized method of claim 1, wherein managing the graphics resource comprises:

determining if the graphics resource is available; and

paging current data associated with the graphics resource to a backing store if the graphics resource is not available.

3. (Original) The computerized method of claim 2, wherein the current data to page is determining using paging criteria.

4. (Original) The computerized method of claim 3, wherein the paging criteria is selected from the group consisting of a type of graphics resource, a priority, and a paging algorithm.

5. (Original) The computerized method of claim 4, wherein the paging algorithm is based on usage of the graphics resource.

6. (Previously Presented) The computerized method of claim 1, wherein managing the graphics resource comprises:

detecting a conflict if the graphics resource has been reused; and

resolving the conflict.

7. (Original) The computerized method of claim 6, wherein resolving the conflict comprises:

inserting a reference to a graphics hardware semaphore before the command;

paging the current data associated with the graphics resource to the backing store;

paging data for the graphics client from the backing store into the graphics resource; and

clearing the graphics hardware semaphore.

8. (Original) The computerized method of claim 1 further comprising:

recording, by the graphics kernel, information about the graphics resource in an entry in a virtualization map for use in allocating and managing graphics resources.

9. (Original) A machine-readable medium having executable instructions to cause a processing system to perform a method comprising:

receiving, by a graphics kernel, an allocation request for a graphics resource from a graphics client;

allocating, by the graphics kernel, the graphics resource to the graphics client;

returning, by the graphics kernel, an address for the graphics resource to the graphics client;

receiving, by the graphics kernel, a command from the graphics client specifying the address; and

managing, by the graphics kernel, the graphics resource.

10. (Original) The machine-readable medium of claim 9, wherein managing the graphics resource comprises:

determining if the graphics resource is available; and

paging current data associated with the graphics resource to a backing store if the graphics resource is not available.

11. (Original) The machine-readable medium of claim 10, wherein the current data to page is determining using paging criteria.

12. (Original) The machine-readable medium of claim 11, wherein the paging criteria is selected from the group consisting of a type of graphics resource, a priority, and a paging algorithm.

13. (Original) The machine-readable medium of claim 12, wherein the paging algorithm is based on usage of the graphics resource.

14. (Original) The machine-readable medium of claim 9, wherein managing the graphics resource comprise:

detecting a conflict if the graphics resource has been reused; and

resolving the conflict.

15. (Original) The machine-readable medium of claim 14, wherein resolving the conflict comprises:

inserting a reference to a graphics hardware semaphore before the command;

paging the current data associated with the graphics resource to the backing store;

paging data for the graphics client from the backing store into the graphics resource; and

clearing the graphics hardware semaphore.

16. (Original) The machine-readable medium of claim 9, wherein the method further comprises:

recording, by the graphics kernel, information about the graphics resource in an entry in a virtualization map for use in allocating and managing graphics resources.

17. (Original) A processing system comprising:

a processor coupled to a memory through a bus;

a graphics processor coupled to the processor through the bus and associated with graphics resources; and

a kernel driver executed by the graphics processor to cause the graphics processor to

receive an allocation request for a graphics resource from a graphics client,

allocate the graphics resource to the graphics client,

return an address for the graphics resource to the graphics client,

receive a command from the graphics client specifying the address, and

manage the graphics resource.

18. (Original) The processing system of claim 17, wherein the kernel driver, when managing the graphics resource, further causes the graphics processor to

determine if the graphics resource is available, and

page current data associated with the graphics resource to a backing store if the graphics resource is not available.

19. (Original) The processing system of claim 18, wherein the current data to page is determined using paging criteria.

20. (Original) The processing system of claim 19, wherein the paging criteria is selected from the group consisting of a type of graphics resource, a priority, and a paging algorithm.

21. (Original) The processing system of claim 20, wherein the paging algorithm is based on usage of the graphics resource.

22. (Original) The processing system of claim 17, wherein the kernel driver, when managing the graphics resource, further causes the graphics processor to

detect a conflict if the graphics resource has been reused; and

resolve the conflict.

23. (Original) The processing system of claim 22, wherein the graphics processor is associated with a graphics hardware semaphore, and the kernel driver, when resolving the conflict, further causes the graphics processor to

insert a reference to the graphics hardware semaphore before the command,

page the current data associated with the graphics resource to the backing store,

page data for the graphics client from the backing store into the graphics resource, and

clear the graphics hardware semaphore.

24. (Original) The processing system of claim 17, wherein the kernel driver further causes the graphics processor to record information about the graphics resource in an entry in a virtualization map for use in allocating and managing graphics resources.

25. (Original) A graphics system comprising:

a graphics processor associated with graphics resources; and

a kernel driver executed by the graphics processor to cause the graphics processor to

receive an allocation request for a graphics resource from a graphics client,

allocate the graphics resource to the graphics client,

return an address for the graphics resource to the graphics client,

receive a command from the graphics client specifying the address, and

manage the graphics resource.

26. (Original) The graphics system of claim 25, wherein the kernel driver, when managing the graphics resource, further causes the graphics processor to

determine if the graphics resource is available, and

page current data associated with the graphics resource to a backing store if the graphics resource is not available.

27. (Original) The graphics system of claim 26, wherein the current data to page is determining using paging criteria.

28. (Original) The graphics system of claim 27, wherein the paging criteria is selected from the group consisting of a type of graphics resource, a priority, and a paging algorithm.

29. (Original) The graphics system of claim 28, wherein the paging algorithm is based on usage of the graphics resource.

30. (Original) The graphics system of claim 25, wherein the kernel driver, when managing the graphics resource, further causes the graphics processor to

detect a conflict if the graphics resource has been reused; and

resolve the conflict.

31. (Original) The graphics system of claim 30, wherein the graphics processor is associated with a graphics hardware semaphore, and the kernel driver, when resolving the conflict, further causes the graphics processor to

insert a reference to the graphics hardware semaphore before the command,

page the current data associated with the graphics resource to the backing store,

page data for the graphics client from the backing store into the graphics resource, and

clear the graphics hardware semaphore.

32. (Original) The graphics system of claim 25, wherein the kernel driver further causes the graphics processor to record information about the graphics resource in an entry in a virtualization map for use in allocating and managing graphics resources.

33. (Original) An apparatus for virtualizing graphics resources comprising:

means for receiving, by a graphics kernel, an allocation request for a graphics resource from a graphics client;

means for allocating, by the graphics kernel, the graphics resource to the graphics client;

means for returning, by the graphics kernel, an address for the graphics resource to the graphics client;

means for receiving, by the graphics kernel, a command from the graphics client specifying the address; and

means for managing, by the graphics kernel, the graphics resource.

34. (Original) The apparatus of claim 33, wherein the means for managing the graphics resource comprises:

means for determining if the graphics resource is available; and

means for paging current data associated with the graphics resource to a backing store if the graphics resource is not available.

35. (Original) The apparatus of claim 34, wherein the means for allocating the graphics resource further comprises:

means for determining the current data to page using paging criteria.

36. (Original) The apparatus of claim 35, wherein the paging criteria is selected from the group consisting of a type of graphics resource, a priority, and a paging algorithm.

37. (Original) The apparatus of claim 36, wherein the paging algorithm is based on usage of the graphics resource.

38. (Previously Presented) The apparatus of claim 33, wherein the means for managing the graphics resource comprises:

means for detecting a conflict if the graphics resource has been reused; and

means for resolving the conflict.

39. (Original) The apparatus of claim 38, wherein the means for resolving the conflict comprises:

means for inserting a reference to a graphics hardware semaphore before the command;

means for paging the current data associated with the graphics resource to the backing store;

means for paging data for the graphics client from the backing store into the graphics resource; and

means for clearing the graphics hardware semaphore.

40. (Original) The apparatus of claim 33 further comprising:

means for recording, by the graphics kernel, information about the graphics resource in an entry in a virtualization map for use by the means for allocating and the means for managing the graphics resource.

41. (New) A computerized method of virtualizing graphics resources comprising:

receiving, by a graphics kernel, an allocation request for a graphics resource from a graphics client;

allocating, by the graphics kernel, the graphics resource to the graphics client;

returning, by the graphics kernel, an address for the graphics resource to the graphics client;

receiving, by the graphics kernel, a command from the graphics client specifying the address;

managing, by the graphics kernel, the graphics resource by detecting a conflict if the graphics resource has been reused; and

resolving, by the graphics kernel, the conflict by inserting a reference to a graphics hardware semaphore before the command, paging the current data associated with the graphics resource to the backing store, paging data for the graphics client from the backing store into the graphics resource, and clearing the graphics hardware semaphore.

42. (New) A machine-readable medium having executable instructions to cause a processing system to perform a method comprising:

receiving, by a graphics kernel, an allocation request for a graphics resource from a graphics client;

allocating, by the graphics kernel, the graphics resource to the graphics client;

returning, by the graphics kernel, an address for the graphics resource to the graphics client;

receiving, by the graphics kernel, a command from the graphics client specifying the address;

managing, by the graphics kernel, the graphics resource by detecting a conflict if the graphics resource has been reused; and

resolving, by the graphics kernel, the conflict by inserting a reference to a graphics hardware semaphore before the command, paging the current data associated with the graphics resource to the backing store, paging data for the graphics client from the backing store into the graphics resource, and clearing the graphics hardware semaphore.

43. (New) A processing system comprising:

a processor coupled to a memory through a bus;

a graphics processor coupled to the processor through the bus and associated with graphics resources; and

a kernel driver executed by the graphics processor to cause the graphics processor to

receive an allocation request for a graphics resource from a graphics client,

allocate the graphics resource to the graphics client,

return an address for the graphics resource to the graphics client,

receive a command from the graphics client specifying the address,

detect a conflict if the graphics resource has been reused, and

insert a reference to the graphics hardware semaphore before the command, page the current data associated with the graphics resource to the backing store, page data for the graphics client from the backing store into the graphics resource, and clear the graphics hardware semaphore to resolve the conflict.

44. (New) A graphics system comprising:

a graphics processor associated with graphics resources; and

a kernel driver executed by the graphics processor to cause the graphics processor to

receive an allocation request for a graphics resource from a graphics client,

allocate the graphics resource to the graphics client,

return an address for the graphics resource to the graphics client,

receive a command from the graphics client specifying the address,

detect a conflict if the graphics resource has been reused, and

insert a reference to the graphics hardware semaphore before the command, page the current data associated with the graphics resource to the backing store, page data for the graphics client from the backing store into the graphics resource, and clear the graphics hardware semaphore to resolve the conflict.

45. (New) An apparatus for virtualizing graphics resources comprising:

means for receiving, by a graphics kernel, an allocation request for a graphics resource from a graphics client;

means for allocating, by the graphics kernel, the graphics resource to the graphics client;

means for returning, by the graphics kernel, an address for the graphics resource to the graphics client;

means for receiving, by the graphics kernel, a command from the graphics client specifying the address;

means for detecting a conflict if the graphics resource has been reused; and

means for inserting a reference to a graphics hardware semaphore before the command;

means for paging the current data associated with the graphics resource to the backing store;

means for paging data for the graphics client from the backing store into the graphics resource; and

means for clearing the graphics hardware semaphore to resolve the conflict.

46. (New) A method of a graphics kernel, comprising:

detecting a conflict if a graphics resource has been reused;

resolving the conflict by inserting a reference to a graphics hardware semaphore before a command received from a graphics client; and

interleaving the processing of a set of command buffers from different graphics clients using the graphics hardware semaphore.